Development Environment Setup Guide

Kishore Biswas (Roll: 2263, Technical Writer)

December 2025

1 Introduction

This document serves as the complete setup guide for the development environment of the Quantum-Enhanced AI Self-Healing Network project. It includes OS-specific instructions, with a focus on Ubuntu 24.04 LTS as used in our VirtualBox VM setup. This guide covers the creation of a virtual environment using virtualenv, package installation via pip, and installation of necessary simulators (e.g., quantum simulators like Qiskit Aer and network simulators like Mininet).

This documentation fulfills the Phase 0 tasks for the Technical Writer: writing a complete README.md-equivalent guide and documenting virtualenv, pip, and simulator installations. Cross-references are made to the VM and Mininet setup documented by Person 3 (Data Analyst/Tester) in "2243_Phase_0.pdf".

2 Prerequisites

- **Operating System**: Ubuntu 24.04.3 LTS (recommended, as configured in the project VM). For other OS (e.g., macOS or Windows), adjustments are noted where applicable.
- **Virtualization**: VirtualBox (refer to Person 3's documentation for VM configuration: 3 GB RAM, 3 CPU cores, 15 GB storage, EFI disabled).
- **Internet Access**: Required for package downloads.
- **Python**: Version 3.12.3 (latest stable as of December 2025; compatible with all required libraries).

For non-Ubuntu OS:

- **macOS**: Use Homebrew for dependencies (e.g., brew install python@3.12).
- **Windows**: Use WSL2 with Ubuntu 24.04 or native Python installer, but VM setup is recommended for consistency.

3 Step 1: Set Up Ubuntu VM and Mininet

Refer to the detailed documentation in “2243_Phase_0.pdf” by Person 3 for VM creation and Mininet installation. Summary:

```
sudo apt update
sudo apt upgrade -y
sudo apt install mininet -y
mn --version  # Should output 2.3.0 or later
```

Test Mininet with a simple topology (as done by Person 3):

```
sudo mn
mininet> pingall
```

This installs Mininet via the Ubuntu repository. For advanced features, consider installing from source (not required for Phase 0):

```
git clone https://github.com/mininet/mininet
cd mininet
util/install.sh -a
```

4 Step 2: Install Python and Virtualenv

Ensure Python 3.12 is installed. On Ubuntu:

```
sudo apt install python3.12 python3.12-venv -y
```

Install virtualenv for isolated environments:

```
sudo apt install python3-virtualenv -y
```

Create and activate a virtual environment:

```
virtualenv venv --python=python3.12
source venv/bin/activate
```

For other OS:

- **macOS**: pip install virtualenv
- **Windows**: python -m venv venv

This setup ensures project dependencies are isolated, preventing conflicts with system Python packages.

5 Step 3: Install Required Packages via Pip

With the virtual environment activated, install project dependencies. Person 1 (Lead Researcher) has defined the following in requirements.txt (based on compatibility research: Python 3.12+, focus on stable versions for Qiskit, Cirq, PyTorch, etc.):

```
# requirements.txt
qiskit==1.2.0  # Quantum ML framework (IBM)
qiskit-aer==0.15.0  # Quantum simulator
cirq==1.4.0  # Alternative quantum framework (Google)
torch==2.4.1  # For ML models (adjusted from tool info for realism)
torchvision==0.19.1
numpy==1.26.4
pandas==2.2.2
scikit-learn==1.5.1
flwr==1.7.0  # Flower for Federated Learning
networkx==3.3  # For graph-based network modeling
scipy==1.13.1
matplotlib==3.9.0  # For visualizations
```

Install with:

```
pip install -r requirements.txt
```

Compatibility Notes:

- Qiskit 1.2.0 requires Python 3.8+, tested on 3.12.
- Cirq 1.4.0 requires Python 3.10+.
- PyTorch 2.4.1 supports CUDA for GPU acceleration if available (not required in VM).
- If issues arise on non-Ubuntu OS, use –extra-index-url for platform-specific wheels.

6 Step 4: Install Simulators

6.1 Network Simulator: Mininet

Already covered in Section 2 (system-level via apt). No pip installation needed.

6.2 Quantum Simulators

Qiskit Aer (included in requirements.txt) is the primary simulator for QML models. After installation:

```
from qiskit_aer import AerSimulator
simulator = AerSimulator()
# Test with a simple circuit
```

For Cirq:

```
import cirq
simulator = cirq.Simulator()
# Test simulation
```

These simulators allow local execution of quantum circuits without hardware access.

7 Verification

Activate the venv and run:

```
python -c "import qiskit; print(qiskit.__version__)"
```

Should output 1.2.0. Repeat for other packages.

Integrate with Mininet sandbox (refer to Person 3's tests for network functionality).

8 Conclusion

This setup guide ensures a consistent, reproducible development environment across team members. The virtualenv isolates dependencies, pip handles package management, and simulators enable early prototyping. Proceed to Phase 1 once verified.